Device Class Power Management Reference Specification

PC Card Controller Device Class

Draft proposal
v0.0

The PC Card Controller Device-class Power Management Specification, including any future enhancements (PC Card-class Specification"), is being provided by Microsoft to encourage the development of devices which exhibit consistent behavior in a Power-managed PC system consistent with the OnNow Design Initiative. Microsoft grants you the right to reproduce, use and distribute the PC Card-class Specification for the purpose of creating, using and distributing hardware which complies with the PC Card-class Specification. Microsoft makes no warrantee that implementations that comply with the PC Card-class specification do not infringe IP rights of third parties.

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted.

Table of Contents

Revision History

Revision	Date	Comments
0.0	3/25/96	Initial proposal for consideration

Scope

This specification defines the behavior of PC Card Controller devices as it relates to power management, and, specifically, to the four device power states defined for the OnNow Architecture. The specification covers the PC Card controller and the PC Card slots specifically. It is intended that PC Card Controller vendors and system makers will be able to design consistent power-manageable products, and that OS vendors will be able to implement an appropriate PC Card Controller power management policy based on the contents of this specification.

General Device Power Management Considerations

In the OnNow architecture, power management of individual devices is the responsibility of a policy owner in the Operating System, generally a class-specific driver. This policy-owner will implement a power conservation policy that is appropriate for devices in its class. The policy will operate in conjunction with a global system power policy implemented in the operating system (i.e. is the system Working or Sleeping?). In general, the device-class power conservation policy strives to reduce power consumption while the system is Working by transitioning amongst various available power states according to device usage. Since the policy-owner in the Operating System has very specific knowledge of when a device is in use, or potentially in use, there is no need for hardware timers or such to determine when to make these transitions. Similarly, this level of understanding of device usage makes it possible to use fewer device power states. Generally, intermediate states attempt to draw a compromise between latency and consumption due to the uncertainty of actual device usage. With the increased knowledge in the OS, crisp decisions can be made about whether the device is needed at all. With this ability to turn devices off more frequently, the benefit of having intermediate states diminishes.

The policy-owner also determines what class-specific events can cause the system to transition from Sleeping to Working, and enables this functionality based on application or user requests. Note that the definition of the wake-up events that each class supports will influence the system's global power policy in terms of the level of power conservation the Sleeping state can attain while still meeting wake-up latency requirements set by applications or the user.

In the OnNow architecture, bus drivers also implement power policy for their bus class (e.g. PCI, USB, etc..). In general, the Bus driver has responsibility for tracking the device power states of all devices on its bus, and transitioning the Bus itself to only those power states that are consistent with those of its devices. This means that the Bus state can be no lower than the highest state of one of it's devices. However, enabled wake-up events can affect this as well. For example if a particular device is in the D2 state and set to wake-up the system, and the bus can only forward wake-up requests while in the D1 state, then the Bus must remain in the D1 state even if all devices are in a lower state.

Device power state transitions are explicitly commanded by the driver and invoked through bus-specific mechanisms (e.g. ATA Standby command, USB Suspend, etc..). Note that the explicit command for entering the D3 state may be the removal of power. In some cases, bus-specific mechanisms are not available and device-specific mechanisms must be used.

The following definitions apply to devices of all classes:

- **D0**: Device is on and running. It is receiving full power from the system, and is delivering full functionality to the user.
- **D1**: Class-specific low-power state (defined below) in which device context may or may not be lost. Buses in D1 cannot do anything to the bus which would force devices on that bus to loose context.
- **D2**: Class-specific low-power state (defined below) in which device context may or may not be lost. Attains greater power savings than D1. Buses in D2 may cause devices on that bus to loose some context (e.g. the bus reduces power supplied to the bus). Devices in D2 must be prepared for the bus to be in D2 (or higher).
- **D3**: Device is off and not running. Device context is lost. Power may be removed from the device.

Any device context lost must be restored by the device driver when returning the device to the D0 state.

PC Card Controller Device State Definitions

D0

- Card status change (from slot): Detectable
- Controller context (e.g. memory, I/O windows): Programmed
- Controller interface: Fully functional (processor can access cards)
- Interrupts to processor: Functional
- Clock to controller: Active
- Power to cards (slots): Available

D1

- This state is not defined for PC Card Controllers. Use D2 instead.

D2

- Card status change (from slot): Detectable
- Controller context (e.g. memory, I/O windows): Preserved
- Controller interface: Non-functional (processor cannot access cards)
- Interrupts to processor: Non-functional
- Clock to controller: Off
- Power to cards (slots): Available

D3 (Power may be removed)

- Card status change (from slot): Not detectable
- Controller context (e.g. memory, I/O windows): Lost
- Controller interface: Non-functional (processor can not access cards)
- Interrupts to processor: Non-functional
- Clock to controller: Off
- Power to cards (slots): Off (card context lost)

PC Card Controller Device Power Conservation Policy

The PC-Card controller is a bus controller. As such, its power state is dependent on the devices plugged into the bus (child devices). The OS will track the state of all devices on the bus and will put the bus into the best possible power state based on the current device requirements on that bus. For example, if the PC-Card cards are all in the D2 state, the OS will put the PC-Card controller in the D2 state.

Present State	Next State	Cause
D1, D2,D3	D0	Any card in any slot needing to transition to state D0
D0	D1	All cards in all slots are in state D1, D2 or D3
D0	D2	All cards in all slots are in state D2 or D3
D0	D3	All cards in all slots are in state D3
D1	D2	All cards in all slots are in state D2 or D3
D1	D3	All cards in all slots are in state D3
D2	D3	All cards in all slots are in state D3

PC Card Controller Device Wake-up Events

Assertion of the controller’s status change line will signal a wake-up event.

Minimum PC Card Controller Device Power Capabilities

TBD